Google Research

Paper Code

Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent

Jaehoon Lee*, Lechao Xiao*, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, Jeffrey Pennington

Google Research, *Equal contribution, work done as part of the Google AI Residency, g.co/airesidency

NeurIPS 2019; arXiv: 1902.06720

Are Training Dynamics Tractable?

A longstanding goal in deep learning research has been to precisely characterize **training** and **generalization**. However, the often complex loss landscapes of neural networks have made a theory of learning dynamics elusive. Nevertheless, in the **large width** limit, neural networks evolve as linear models with solvable dynamics.

a. **Parameter space dynamics:** wide network training dynamics in parameter space are equivalent to the training dynamics of a model which is affine in the collection of all network parameters.
b. **Sufficient conditions for linearization:** there exists a threshold learning rate $\eta_{critical}$, such that gradient descent training of neural networks with learning rate smaller than that threshold are well approximated by their linearization for large width.
c. **Output distribution dynamics:** the predictions of a neural network throughout gradient descent training converge weakly to a GP as the width goes to infinity. We derive time-dependent expressions for the evolution of this GP and note the differences from the Bayesian posterior GP.
d. **More in the paper:**
   - Parameterization independence for linearization
   - Momentum, non-square losses (e.g. cross-entropy)
   - Analytic expressions for NTK for Erf and Relu
   - Colab tutorial

Motivation: Gradient descent learning dynamics of deep neural networks are intractable

- **Consider (convex) loss function of neural networks** $\mathcal{L} = \sum_{(x,y)\in\mathcal{D}} \ell(f_t(x,\theta), y).$
- **Gradient descent (flow) dynamics of the parameters and outputs (logits)**

$$\dot\theta_t = -\eta \nabla_\theta f_t(\mathcal{X})^T \nabla_{f_t(\mathcal{X})}\mathcal{L}$$
$$\dot f_t(\mathcal{X}) = \nabla_\theta f_t(\mathcal{X})\,\dot\theta_t = -\eta\,\hat\Theta_t(\mathcal{X},\mathcal{X})\nabla_{f_t(\mathcal{X})}\mathcal{L}$$

- **With a time evolving tangent kernel** $\hat\Theta_t = \nabla_\theta f_t(\mathcal{X})\nabla_\theta f_t(\mathcal{X})^T$

Dynamics of linear models are tractable

- **For linear models, training dynamics are tractable. Closed form solution for MSE loss:**

$$\omega_t = -\nabla_\theta f_0(\mathcal{X})^T\hat\Theta_0^{-1}(I - e^{-\eta\hat\Theta_0 t})(f_0(\mathcal{X}) - \mathcal{Y}) \qquad \text{weight dynamics}$$
$$f_t^{\rm lin}(\mathcal{X}) = (I - e^{-\eta\hat\Theta_0 t})\mathcal{Y} + e^{-\eta\hat\Theta_0 t} f_0(\mathcal{X}) \qquad \text{function dynamics}$$

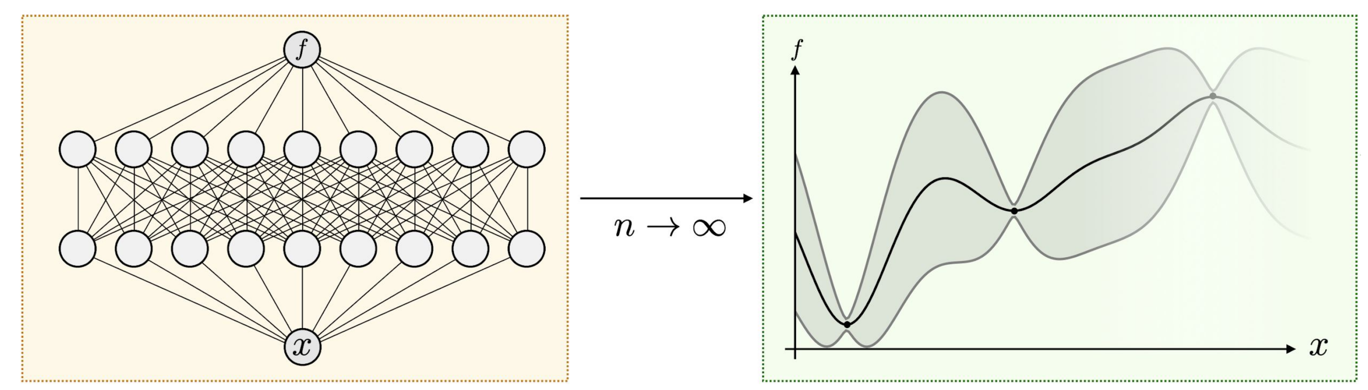
- **For some test point x** $f_t^{\rm lin}(x) = \mu_t(x) + \gamma_t(x)$

$$\mu_t(x) = \hat\Theta_0(x,\mathcal{X})\hat\Theta_0^{-1}(I - e^{-\eta\hat\Theta_0 t})\mathcal{Y} \qquad \text{deterministic}$$
$$\gamma_t(x) = f_0(x) - \hat\Theta_0(x,\mathcal{X})\hat\Theta_0^{-1}(I - e^{-\eta\hat\Theta_0 t})f_0(\mathcal{X}) \qquad \text{stochastic}$$

Infinite networks are GPs

As the width approaches infinity, the outputs of randomly initialized network converge to a Gaussian Process [1-3]:

$$f_0(\mathcal{X}) \sim \mathcal{N}(0, \mathcal{K}(\mathcal{X},\mathcal{X})) \qquad \mathcal{K}^{i,j}(x,x') = \underset{n\to\infty}{\text{p-lim}}\,\mathbb{E}\left[f_0^i(x) f_0^j(x')\right]$$



This GP is transformed by gradient descent throughout the training process, leading to GP behavior after training.

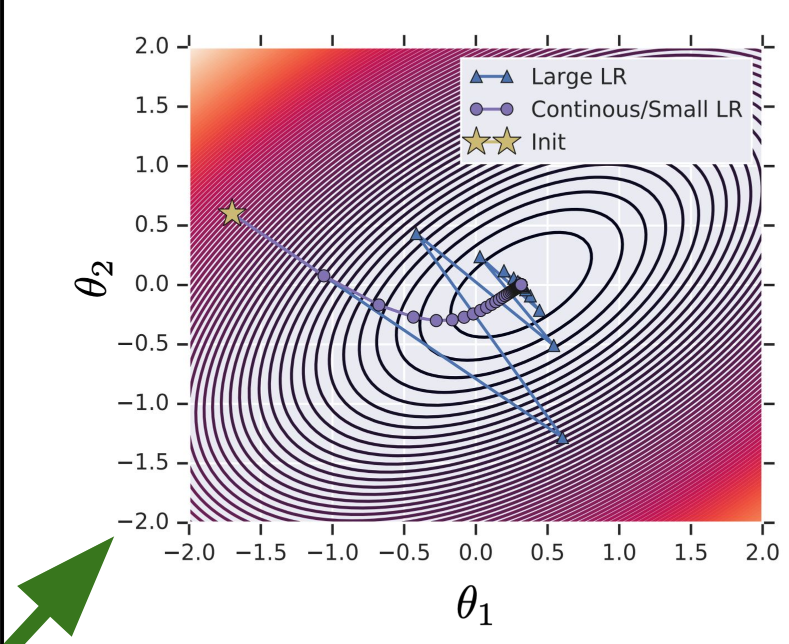
Infinite networks are linearized models

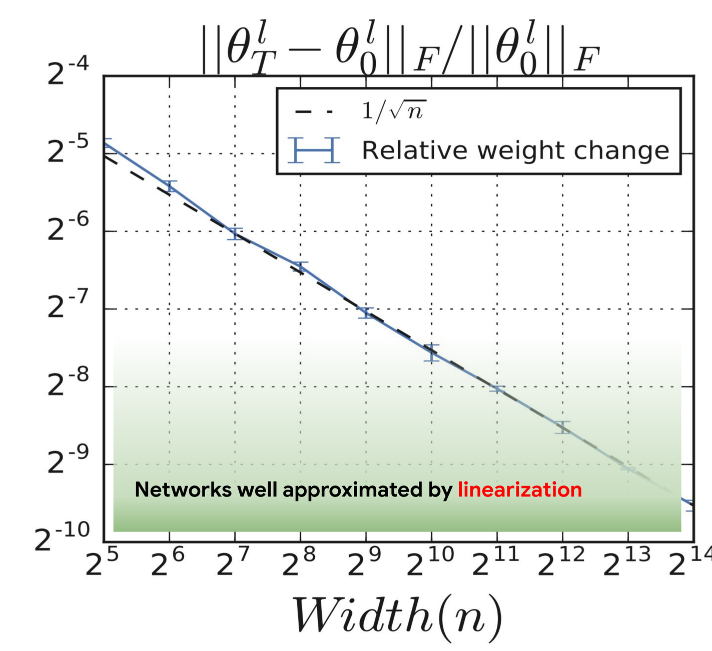
- **When the network is sufficiently wide, one can approximate a deep neural network by its first order Taylor expansion (linearization) at initialization (t=0)**

$$\boxed{f_t^{\rm lin}(x) \equiv f_0(x) + \nabla_\theta f_0(x)\,\omega_t} \qquad \omega_t \equiv \theta_t - \theta_0$$

- **When learning rate is sufficiently small, linearization becomes more accurate as the width increases:**

$$\sup_{t\geq 0}\left\|f_t(x) - f_t^{lin}(x)\right\|_2,\ \sup_{t\geq 0}\frac{\|\theta_t - \theta_0\|_2}{\sqrt{n}},\ \sup_{t\geq 0}\left\|\hat\Theta_t - \hat\Theta_0\right\|_F = \mathcal{O}(n^{-\frac{1}{2}}),\ \text{as}\quad n\to\infty$$

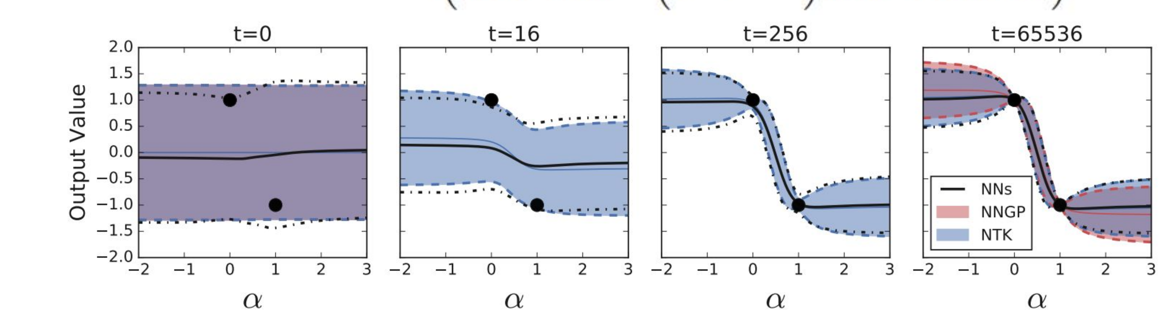




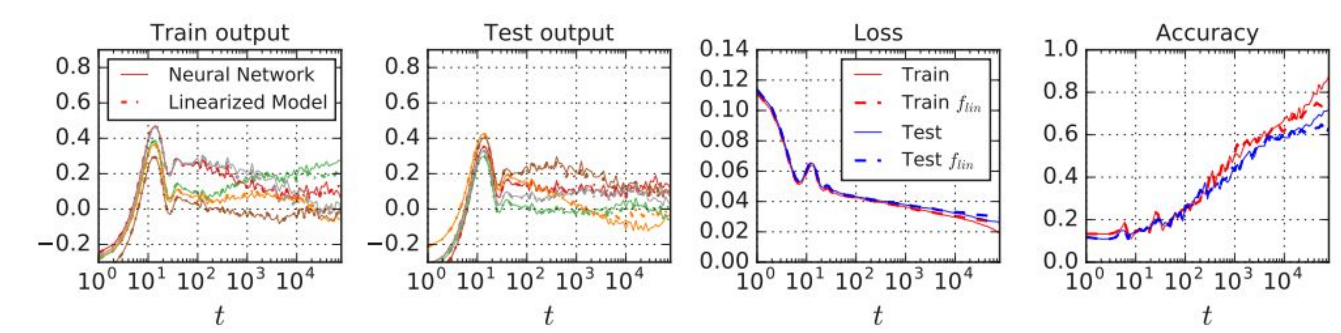
Experiments

- **Gaussian process at initialization leads to Gaussian distribution during training with mean and covariance**

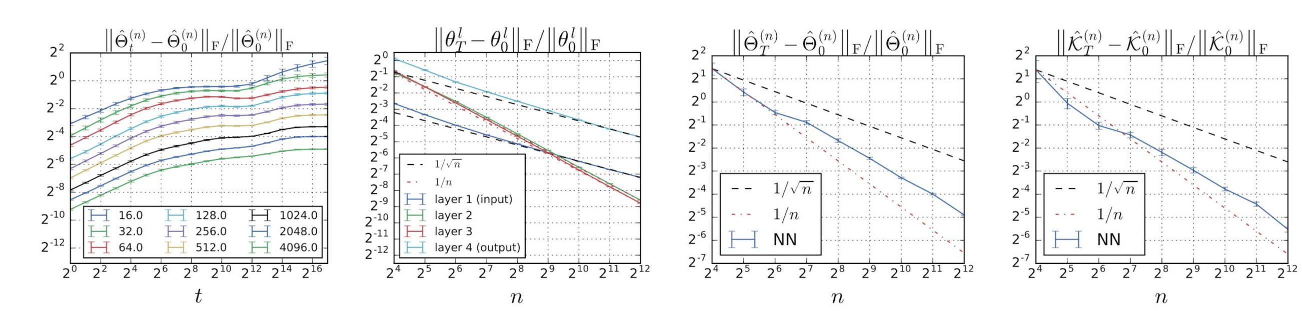
$$\mu(\mathcal{X}_T) = \Theta(\mathcal{X}_T,\mathcal{X})\Theta^{-1}\left(I - e^{-\eta\Theta t}\right)\mathcal{Y},$$
$$\Sigma(\mathcal{X}_T,\mathcal{X}_T) = \mathcal{K}(\mathcal{X}_T,\mathcal{X}_T) + \Theta(\mathcal{X}_T,\mathcal{X})\Theta^{-1}\left(I - e^{-\eta\Theta t}\right)\mathcal{K}\left(I - e^{-\eta\Theta t}\right)\Theta^{-1}\Theta(\mathcal{X},\mathcal{X}_T)$$
$$- \left(\Theta(\mathcal{X}_T,\mathcal{X})\Theta^{-1}\left(I - e^{-\eta\Theta t}\right)\mathcal{K}(\mathcal{X},\mathcal{X}_T) + h.c.\right).$$



- **A wide residual network and its linearization behave similarly when both are trained by SGD with momentum on MSE loss on full CIFAR-10**



- **Relative Frobenius norm change during training**



Relation to other approaches

	Weight Space	Function Space
Bayesian Training	$p(z^*\|x^*,\mathcal{D}) = \int d\theta\, p(z^*\|x^*,\theta)\, p(\theta\|\mathcal{D})$ **Bayesian Neural Network**	$p(z) = \mathcal{GP}(z; 0, K(x))$ $z(x^*)\|\mathcal{D} \sim \mathcal{GP}(z_t(x^*)\|\mathcal{D})$ **NNGP**
Gradient Descent	$z(x;\theta_t) = z(x;\theta_0) + \nabla_\theta z(x;\theta_0)(\theta_t - \theta_0) + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ $\partial_t\theta_t = -\eta(\nabla_\theta z_t)^T\nabla_{z_t}\mathcal{L}(z_t(x))$ **Linearization**	$\partial_t z_t(x) = -\eta\hat\Theta\nabla_{z_t}\mathcal{L}(z_t(x))$ $z(x^*)\|\mathcal{D} \sim \mathcal{GP}(z_t(x^*); \mu_t(\mathcal{D}), \Sigma_t(\mathcal{D}))$ **Neural Tangent Kernel**

References

1. Neal, Radford M. "Bayesian learning for neural networks", Ph.D. Thesis, University of Toronto, 1994.
2. Lee, J. and Bahri, Y., et al. "Deep neural networks as gaussian processes", *arXiv:1711.00165*, ICLR 2018.
3. Matthews, Alexander G. de G., et al. "Gaussian process behaviour in wide deep neural networks", *arXiv:1804.11271*, ICLR 2018.
4. Jacot, A. , et al. "Neural Tangent Kernel: Convergence and Generalization in Neural Networks", *arXiv:1806.07572*, NeurIPS 2018.
5. Novak R. and Xiao L., et al. "Neural Tangents: Fast and Easy Infinite Neural Networks in Python", arXiv:1912.02803.

Want to automate NNGP/NTK computation for various architectures?
: see "***Neural Tangents***" github.com/google/neural-tangents [5]!